PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 1/02	A1	(11) International Publication Number: WO 91/10182
		(43) International Publication Date: 11 July 1991 (11.07.91)

(21) International Application Number: PCT/US90/04407

(22) International Filing Date: 7 August 1990 (07.08.90)

(30) Priority data:
454,763 21 December 1989 (21.12.89) US

(71) Applicant: BELL COMMUNICATIONS RESEARCH, INC. [US/US]; 290 West Mount Pleasant Avenue, Livingston, NJ 07039-2729 (US).

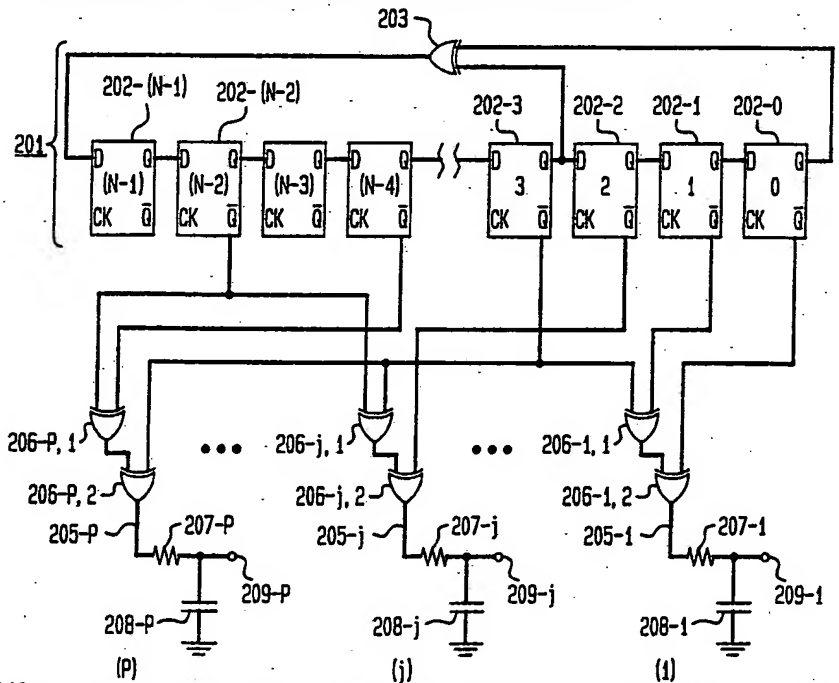
(72) Inventors: ALSPECTOR, Joshua ; 716 Belvidere Avenue, Westfield, NJ 07090 (US). CHU, Robert, Ray ; 38075 Geranium Street, Newark, CA 94560 (US). GANNETT, Joel, Wright ; 9 Conklin Avenue, Morristown, NJ 07960 (US). HABER, Stuart, Alan ; 22 Irving Place, Apt. 2C, Manhattan, NY 10003 (US). PARKER, Michael, Benjamin ; 721 East Walnut Avenue, Orange, CA 92667-6833 (US).

(74) Agents: SUCHYTA, Leonard, Charles; Pct International, Inc., International Coordinator, Room 2E-304, Bell Communications Research, Inc., 290 West MOunt Pleasant Avenue, Livingston, NJ 07039 (US) et al.

(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent)*, DK (European patent), ES (European patent), FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.

(54) Title: GENERATOR OF MULTIPLE UNCORRELATED NOISE SOURCES



(57) Abstract

Plural, arbitrarily-shifted, pseudo-random bits streams are generated from a single linear feedback shift register (LFSR) (201). Each bit stream is obtained by tapping the outputs of selected LFSR cells (202) and feeding these tapped cell outputs through a set of exclusive-OR gates (206). The taps are selected in order to achieve the desired shift between bit streams. In addition, the tap patterns can be selected so that the number of inputs (fan-in) to each bit stream are within predetermined bounds and that the number of taps per cell (cell load) are within predetermined bounds. A disclosed computer program generates the tap patterns as a function of the number of cells and the structure of the LFSR, the number of output bit streams, the maximum allowed shift variation of the bit streams, and the bounds on fan-in and cell load. Each pseudo-random bit stream serves as an input to a low-pass filter which produces an essentially Gaussian noise output. The plural noise outputs are relatively uncorrelated and can be used in a parallel stochastic learning neural network for purposes such as annealing.

DESIGNATIONS OF "DE"

Until further notice, any designation of "DE" in any international application whose international filing date is prior to October 3, 1990, shall have effect in the territory of the Federal Republic of Germany with the exception of the territory of the former German Democratic Republic.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		

GENERATOR OF MULTIPLE UNCORRELATED NOISE SOURCES

BACKGROUND OF THE INVENTION

In a prior art neural network test chip, a stochastic learning technique with a local learning rule was implemented in VLSI. (see, for example, U.S. Patent No. 4,874,964, issued October 17, 1989 to J. Alspector and R. B. Allen; J. Alspector and R. B. Allen, "A neuromorphic vlsi learning system," in *Advanced Research in VLSI: Proceedings of the 1987 Stanford Conference*, P. Losleben, Ed. Cambridge, MA: MIT Press, pp. 313-349, 1987; J. Alspector, R. B. Allen, V. Hu, and S. Satyanarayanna, "Stochastic learning networks and their electronic implementation," Proceedings of the conference on *Neural Information Processing Systems*, Denver, CO, pp. 9-21, Nov. 1987, D. Anderson, Ed. New York, NY: Am. Inst. of Phys., 1988; and J. Alspector, B. Gupta, and R. B. Allen, "Performance of a stochastic learning microchip" in *Advances in Neural Information Processing Systems 1*, Denver, CO, pp. 748-760, November 1988, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989). The Boltzman algorithm (D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science* 9, pp. 147-169, 1985) depends on the stochastic settling of the neural system using the process of simulated annealing (S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, 220, pp. 671-680, 1983) to avoid local minima in the energy function that describes its evolution. In the aforenoted prior art neural network prototype test chip, highly amplified Gaussian thermal noise generated by electrons in a transistor was used for annealing. Each neuron was fed by a separate thermal noise generator, so that its state would be

unaffected by the noise seen by the others.

Neural learning algorithms such as this capture correlations seen by neural states to perform classification based on input data. For local learning rules, stochastic elements are necessary for, among other reasons, performing unbiased averaging over neural states elsewhere in the network. Correlations in the noise they see would cause errors in the learning since these undesired correlations would be captured by the learning rule. Other reasons for stochastic elements in neural networks include the search of a large solution space, helping a network settle while avoiding local minima, and interpolating between discrete values of weights by time averaging.

Although a thermal noise generator seems simple and unbiased it has implementation problems. In particular, it exacts a substantial area penalty; and, in fact, occupies much more area than the neuron itself. More significantly, the large gain needed to amplify thermal noise can lead to cross coupling of the on-chip amplifiers thereby frustrating the original purpose of using separate noise amplifiers to obtain zero cross correlation. Despite this, the small network on the prior art test chip demonstrated satisfactory learning for small problems. To scale this network to larger size, it would have to be sensitive to more subtle correlations and therefore the noise sources must show minimal correlation.

A linear feedback shift register (LFSR) produces a pseudo-random bit stream (PRBS) that can be used to make an analog noise source. The PRBS is processed by a low-pass filter with cutoff frequency just a few percent of the clock frequency. This has the effect of performing a time integration over many bits. If each bit's value is randomly distributed with a probability of 0.5 for 0 or 1, then the value of this integration follows a binomial distribution that approaches a Gaussian distribution for a large number of bits. This creates a Gaussian analog pseudo-random noise source whose statistical properties are similar to the thermal noise which is to be modeled with a simulated annealing technique. Variable amplifiers with gains low enough to avoid coupling problems are then sufficient to perform the annealing process. An N-stage LFSR creates a PRBS of maximal length, $2^N - 1$, when the feedback taps are chosen appropriately. One useful property of such a PRBS is that it has cross correlation $-1/(2^N - 1)$ (effectively negligible) with a time shifted version of itself, assuming the cross correlation is calculated after replacing each 1 of the binary bit stream

with −1 and each 0 with 1. (see, for example, S. W. Golomb, *Shift Register Sequences*, revised ed. Laguna Hills, CA: Aegean Park Press, 1982.) For neural network purposes, this time shift must be large enough for the network to settle sufficiently to "forget" the sequence during the anneal cycle before it sees another version of it later. In practice, this is obtained easily with relatively small shift registers because the length of the sequence grows exponentially with the shift register size.

This shifting could be accomplished by using a collection of identical LFSRs, one per neuron. Each would be loaded with a specified initial state to obtain a desired shift relative to the other LFSRs. All LFSRs would be clocked simultaneously. The overhead of such an approach, however, is unacceptable. For instance, a single 25-stage shift register (with a maximal period of 34 million clock cycles) would require approximately 400,000 square microns in 2 micron CMOS technology, which is considerably larger even than the thermal noise amplifier of the prior art implementation in the same technology.

Various techniques for generating plural PRBS have been reported. For example, P. D. Hortensius, R. D. McLeod, W. Pries, D. M. Miller, and H. C. Card, describe a "Cellular automata-based pseudorandom number generators for built-in self-test," in *IEEE Trans. Computer-Aided Design*, vol. 8, no. 8, pp. 842-859, Aug. 1989. As disclosed therein, cellular automata are employed to generate pseudo-random bits in parallel. W. J. McFarland, K. H. Springer, and C.-S. Yen, describe a "1-gword/s pseudorandom word generator," in *IEEE J. Solid-State Circuits*, vol. 24, no. 3, pp. 747-751, June 1989. This pseudorandom word generator uses a feedback/feedforward technique with exclusive-OR gates at each shift register stage. This technique requires as least as many shift register stages as outputs. A wideband digital pseudo-Gaussian noise generator is disclosed in U.S. Patent No. 3,747,381, issued June 26, 1973 to W. J. Hurd. This noise generator requires at least two feedback shift registers of relatively prime lengths. Disadvantageously, in all these prior art noise and/or PRBS generators, the number of cells required linearly increases with the number of required bit streams, P.

An object of the present invention is reduce to a minimum the hardware necessary to generate multiple pseudo-random noise sources required for annealing in neural networks.

An additional object of the present invention is to amortize the space required for a single generator of plural noise sources amongst many neurons in a neural network so that an acceptable small area overhead for VLSI implementations results.

SUMMARY OF THE INVENTION

In accordance with the present invention a single maximal length linear feedback shift register is used to generate multiple, arbitrarily-shifted, pseudo-random bit streams. Each bit stream is converted to an analog noise source by filtering. In particular, each bit stream is obtained by tapping the outputs of selected LFSR cells and feeding these tapped cell outputs through a parity tree consisting of exclusive-OR gates. In accordance with the invention, the particular cells of the LFSR tapped to form each bit stream are selected to meet certain constraints. In particular the taps are chosen so that: (1) the shift variation between bit streams is within a set limit; (2) each cell is tapped to provide an input to no fewer than and no greater than preset numbers of bit streams; and (3) each bit stream is formed from no fewer than and no greater than preset numbers of cell outputs.

An advantage of the present invention is that the number of cells needed to produce P bit streams grows as $\log(P)$ rather than linearly with P.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 is a schematic diagram of a conventional prior art linear feedback shift register used to make an analog noise source; and

FIG. 2 is a schematic diagram of a single linear feedback shift register used to generate multiple pseudo-random bit streams in accordance with the present invention.

DETAILED DESCRIPTION

With reference to FIG. 1, the single N-stage LFSR 101, also denoted Y in the equations derived hereinbelow, consists of N clocked D-type flip-flops 102-(N-1) - 102-0. The N stages, also called *cells*, are arrayed horizontally with the shift direction from left to right, i.e., the input of every cell except the leftmost cell is connected directly to the output of the cell on its left. The cells are numbered consecutively from $(N-1)$ to 0, with the $(N-1)$th cell, 102-(N-1), on the left and the zeroth cell, 102-0, on the right. The signal fed to the D input of the $(N-1)$th (leftmost) cell, 102-(N-1), is obtained from the *feedback function* H. This is the modulo 2 sum of the outputs belonging to a subset of the N cells, that is,

$$H \triangleq \sum_{i=0}^{N-1} c_i z_i \pmod 2 \qquad (1)$$

where $\triangleq$ denotes "is defined to be equal to," z_i is the output of cell i, and each *feedback coefficient* c_i is either 0 or 1. In the embodiment of FIG. 1, c_0 and c_3 equal 1 and the other c_i equal 0. These are just chosen for illustration and in reality would be determined as a function of N and the primitive polynomial thereof, to be defined hereinbelow. Exclusive-OR gate 103 forms the modulo 2 sum of the two fed back outputs of cells 102-0 and 102-3. The output of gate 103 provides the D-input to cell 102-(N-1).

To *shift* register Y 101 means to apply one or more clock pulses simultaneously to the CK clock inputs cells of Y 101. The clock is not shown. The *PRBS generated by* Y is, by definition, the sequence of bits generated by the zeroth (rightmost) cell, one bit per clock cycle, as Y is shifted. The sequence of states that Y evolves through as it is shifted is determined by the initial state and the feedback function H. Thus, the PRBS for a given LFSR depends on its initial state. If Y sequences through *all* possible nonzero states whenever it starts in a nonzero initial state, Y is said to be *maximal*. Maximality occurs only for certain choices of the

feedback coefficients c_i, namely, if the polynomial $c(x)$, where $c(x)$ is defined by the expression

$$c(x) \triangleq x^N + \sum_{i=0}^{N-1} c_i x^i \qquad (2)$$

is primitive in GF(2^N), where GF(2^N) denotes the Galois field with 2^N elements. A PRBS generated by a maximal N-stage LFSR starting in a nonzero initial state is called an *N-maximal PRBS*. Some straightforward implications of maximality include (a) an N-maximal PRBS has period $2^N - 1$, and (b) every possible combination of N consecutive bits, except the all-zero combination, occurs somewhere in an N-maximal PRBS.

In the prior art analog noise generator in FIG. 1, the pseudo-random bit sequence is taken at the $\overline{Q}$ output of the rightmost cell, 102-0. This digital bit sequence on lead 104 is processed by a low pass filter having a cutoff frequency just a few percent of the clock frequency, and consisting of resistor 105 and capacitor 106.. An essentially Gaussian analog pseudo-random noise source is thus created at output 107.

With reference to FIG. 2, a single maximal length LFSR 201 is used to derive plural pseudo-random bit streams. As in the prior art described hereinabove, LFSR consists of N cells, 202-(N-1) - 202-0. Feedback is provided to the D input of cell 202-(N-1) as determined by a primitive polynomial of the N-stage register. As in the prior art structure, feedback is provided in this illustrative example from the Q outputs of the 0th and 3rd cells, 202-0 and 202-3, respectively, which are modulo 2 summed by exclusive-OR gate 203. As above, these particular cells are selected just for purposes of illustration.

It has been determined and mathematically proven by the inventors herein, that by tapping and modulo 2 combining the outputs of particularly selected cells of the maximal length LFSR, shifted versions of the basic bit pattern can be generated. By proper selection of the cells tapped, in fact, any of the $2^N - 1$ possible shifts can be generated. If the shifts are sufficiently far apart, each combination of cell outputs can serve as a separate source of noise that is essentially uncorrelated with the other sources generated from the same LFSR. It is thus necessary to know which cells to tap to generate the plural bit streams that are shifted sufficiently apart to ensure low correlation. As will be described, the cells tapped can be chosen such that in addition to meeting a shift constraint, other constraints can be met that affect the physicality of a VLSI implementation. Advantageously, the number of bit streams that can be generated from the

single LFSR is not limited to the number of cells in the shift register.

In the purely illustrative example in FIG.2, P sources of random Gaussian noise are generated. As just noted, these noise sources are generated from P pseudo-random bit streams, which are shifted versions of each other, by modulo 2 combining the $\overline{Q}$ outputs of selected cells in the register. In the illustrative example of FIG. 2, the first bit stream on lead 205-1 is produced from the modulo 2 combination of the $\overline{Q}$ outputs of cells 202-0, 202-1, and 202-3 which are combined by exclusive-OR gates 206-1,1 and 206-1,2. The bit stream on lead 205-1 is low-pass filtered by the RC filter, consisting of resistor 207-1 and capacitor 208-1, to produce the random noise source on lead 209-1. The other noise sources on leads 205-j, for $2 \leq j \leq P$, are similarly produced by modulo 2 combining, through exclusive-OR gates 206-j,1 and 206-j,2, the outputs of selected of the cells. The resultant bit stream is then filtered through a low-pass filter consisting of resistor 207-j and capacitor 208-j, to produce random noise at output 209-j. In this illustrative example, each output bit stream is generated from three cell outputs. As will be noted hereinafter, the minimum and maximum number of cells needed to be tapped to form any of the bit streams from the LFSR, called the *minimum* and *maximum allowed fan-in*, respectively, is a factor that can be controlled in selecting the tap patterns. Also, the minimum and maximum number of taps on any one cell in the LFSR, called the *minimum* and *maximum allowed cell load*, respectively, is controllable.

In what follows, an algorithm for determining the taps will be provided. First, however, a mathematical foundation will be presented for the technique of the present invention. Two lemmas that are keys to the technique of the present invention for generating multiple bit streams from a single LFSR will be proven. The first lemma loosely says that the bit stream obtained from the modulo 2 combination of the outputs of the cells of a maximal LFSR gives a shifted version of the basic LFSR bit stream. The second lemma says that *any* desired shift can be obtained by appropriate choice of the taps.

Preceding the rigorous mathematical foundation, let A_0 denote an N-maximal PRBS. For each nonnegative integer k, let $a_k \in \{0, 1\}$ denote the value of the sequence A_0 at clock cycle k. This is indicated with the notation $A_0 = \{a_0, a_1, a_2, \cdots\}$. For every positive integer m, define $A_m \triangleq \{a_m, a_{m+1}, a_{m+2}, \cdots\}$. Note that A_m is obtained from A_0 by shifting

forward in time by m clock cycles. Finally, for a given nonzero initial state of Y, let S denote the set containing the all-zero sequence along with the shifted sequences A_m, where $0 \leq m \leq 2^N - 2$. Lemma 1, which says, in general terms, that the bitwise exclusive-OR of two shifted versions of a given N-maximal PRBS generates another shifted version of the same PRBS, can now be stated:

Lemma 1: Let m and n be nonnegative integers. Let $B = \{b_0, b_1, b_2, \cdots\}$ denote the sequence obtained by a bitwise exclusive-OR of $A_m \in S$ and $A_n \in S$, that is, $b_i \triangleq a_{m+i} + a_{n+i} \pmod 2$. Then $B \in S$.

Proof: A_0 is generated by a recursion relation of the form

$$a_{k+N} = \sum_{i=0}^{N-1} c_i a_{k+i} \pmod 2 \qquad (3)$$

where the feedback coefficients c_i are either 0 or 1. Clearly, A_m and A_n also satisfy this recursion relation. Since Eq. (3) is linear, B (which equals the bitwise modulo 2 sum of A_m and A_n) satisfies Eq. (3) as well. Thus, the entire sequence B is determined by its first N bits. Suppose m and n are equal modulo $2^N - 1$. Then A_m and A_n are identical sequences; thus, B is the all-zero sequence and is therefore a member of S. Now suppose m and n are unequal modulo $2^N - 1$. Then A_m and A_n are not identical sequences and B is not the all-zero sequence. In particular, the first N bits of B cannot all be zero (otherwise, Eq. (3) would imply that B is the all-zero sequence). Since A_0 is an N-maximal PRBS, all possible combinations of N consecutive bits except the all-zero combination must occur in A_0. Thus, there must be some nonnegative r such that the first N bits of A_r equal the first N bits of B. Since A_0 is periodic with period $2^N - 1$, there is no loss of generality in assuming $r < 2^N - 1$. Thus $B = A_r \in S$. **Q.E.D.**

Lemma 1 is a special case of the more general *Abelian group* property of S under bitwise modulo 2 addition.

A pair of taps from an LFSR gives two particular shifted sequences from a restricted set. Their exclusive-OR gives a third sequence by Lemma 1. This sequence, in turn, can be exclusive-OR'ed with another tap to give still other shifted versions of the main sequence. Lemma 1 thus implies that given a maximal LFSR generating a PRBS, the

outputs of a collection of cells of this LFSR can be tapped and the mod 2 sum of these outputs taken to obtain a shifted version of the PRBS. The question arises whether any specified shift can be obtained by appropriate choice of the taps. Lemma 2 hereinbelow answers this question in the affirmative.

Lemma 2: Let Y denote an N-stage maximal LFSR that is initialized to a nonzero state, and let z_i^k, $0 \le i \le N-1$, $k \ge 0$, denote the output of cell i of Y at clock cycle k. For a collection of coefficients $d_i \in \{0, 1\}$, $0 \le i \le N-1$, define a sequence $G \triangleq \{g_0, g_1, g_2, \cdots\}$ such that

$$g_k \triangleq \sum_{i=0}^{N-1} d_i z_i^k \pmod 2 \qquad (4)$$

Then for every $A_r \in S$, there exists a collection of coefficients d_i such that $G = A_r$.

Note: In what follows, the coefficients d_i are called the *tap coefficients*. $\mathbf{F}^N$ denotes the set of N-dimensional vectors with components 0 and 1. $\mathbf{F}^N$ can be identified with GF(2^N).

Proof: From Lemma 1, $G \in S$. Each collection of tap coefficients $\mathbf{d} = [d_0, d_1, \cdots, d_{N-1}]^T$ (T denotes transpose) is identified with a member of $\mathbf{F}^N$. Consider the function $Q: \mathbf{F}^N \to S$ that maps (according to Eq. (4)) each tap coefficient vector $\mathbf{d} \in \mathbf{F}^N$ to its corresponding sequence $G \in S$. It will be shown that Q is injective. Since Q is a linear map, it is injective if and only if it maps all nonzero points in its domain to nonzero points in its range. Let $\mathbf{d}^*$ be a nonzero point of $\mathbf{F}^N$. Then there exists an m such that the mth element of $\mathbf{d}^*$ is not zero, that is, $d_m^* = 1$. Since the shift register Y is maximal, there exists a clock cycle k such that $z_m^k = 1$ and $z_i^k = 0$ for all $i \ne m$. By Eq. (4), the bit value of $G^* \triangleq Q(\mathbf{d}^*)$ at clock cycle k is $d_m^* = 1$. Thus, G^* is not the all-zero sequence. Note that $\mathbf{F}^N$ has 2^N elements and S has 2^N sequences. Since the function Q is injective, it follows that Q is surjective because its domain and range have a finite and equal number of elements. **Q.E.D.**

It is thus proven that for a maximal LFSR, the $2^N - 1$ nonzero tap patterns map uniquely to the $2^N - 1$ possible shift values $(0, 1, \cdots, 2^N - 2)$. Therefore, any shift is possible if the right tap pattern is found, and each tap pattern can be identified with a unique shift.

These two viewpoints form the basis for the practical problem to be solved; namely, that of generating properly shifted versions of the original bit stream in a hardware-efficient manner.

The constraints due to VLSI implementation of a neural net model are first described:

1. The bit streams should be shifted far enough apart so that the network can settle without seeing a shifted version of a noise source in two places. This implies close to equal spacing of the bit streams. In practice, this constraint can be relaxed considerably or eased by simply increasing the shift register size.
2. For performance reasons, the fan-out per cell is limited; that is, loading any flip-flop in the register more than is necessary should be avoided.
3. As few inputs as possible to each set of exclusive-OR gates associated with a bit stream is desired. This reduces silicon area and improves performance. In fact, layout simplicity may require an equal number for all sets.

To formulate a precise problem statement, again let $\mathbf{Y}$ denote a maximal N-stage LFSR, and let $p \triangleq 2^N - 1$ denote the period of the PRBS A_0 generated by $\mathbf{Y}$ for some specified nonzero initial state. Let $\underline{L}$ be a nonnegative integer, let $\overline{L}$, $\underline{F}$, $\overline{F}$, and P be positive integers, and let r be a real number such that $0 \le r < 1$. Let $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{P-1} \in \mathbf{F}^N$ denote a collection of P tap coefficient vectors to be determined, and let $G_i \in S$ denote the sequence corresponding to $\mathbf{d}_i$, as in the proof of Lemma 2. Let s_i denote the shift of G_i relative to A_0, where $0 \le s_i < p$. Without loss of generality, assume $s_i \le s_{i+1}$ for all $i < P-1$. Define the *shift differences* t_i, $0 \le i \le P-1$, as follows:

$$t_i \triangleq \begin{cases} s_{i+1} - s_i, & \text{if } 0 \le i \le P-2 \\ p + s_0 - s_{P-1}, & \text{if } i = P-1 \end{cases} \tag{5}$$

Let $\mathbf{u} \in \mathbf{R}^P$ denote the P-vector whose ith component is $u_i = |t_i/(p/P) - 1|$. This is the normalized version of the shift difference. Two vectors, $\mathbf{l} \in \mathbf{R}^N$ and $\mathbf{f} \in \mathbf{R}^P$, both of which have integer-valued components, are associated with a given collection of tap coefficient vectors $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{P-1} \in \mathbf{F}^N$. The component l_i of $\mathbf{l}$ is the number of taps connected to cell i of $\mathbf{Y}$. The component f_i of $\mathbf{f}$ is the number of 1s in (i.e., the number of cell taps

represented by) the tap vector $\mathbf{d}_i$. Let $C: \mathbf{R}^P \times \mathbf{R}^P \times \mathbf{R}^N \rightarrow [0, \infty)$ denote a *cost function*. $C(\mathbf{u}, \mathbf{f}, \mathbf{l})$ is the cost associated with a collection of tap coefficient vectors $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{P-1} \in \mathbf{F}^N$. With these definitions, the problem can be stated precisely. The implementation constraints noted above can be restated in mathematical terms as follows:

Problem Statement: A collection of tap coefficient vectors $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{P-1} \in \mathbf{F}^N$ needs to be found that minimizes the cost $C(\mathbf{u}, \mathbf{f}, \mathbf{l})$ subject to the following conditions:

1. $u_i = |t_i/(p/P) - 1| \leq r$ for all i. The parameter r is the *maximum allowed shift variation*.
2. No cell of Y has fewer than $\underline{L}$ taps or more than $\overline{L}$ taps ($\underline{L} \leq l_i \leq \overline{L}$ for $0 \leq i \leq N-1$). The integer $\underline{L}$ (resp., $\overline{L}$) is the aforenoted minimum (resp., maximum) allowed cell load.
3. No tap coefficient vector $\mathbf{d}_i$ has fewer than $\underline{F}$ components equal to 1 or more than $\overline{F}$ components equal to 1 ($\underline{F} \leq f_i \leq \overline{F}$ for $0 \leq i \leq P-1$). The integer $\underline{F}$ (resp., $\overline{F}$) is the aforenoted minimum (resp., maximum) allowed fan-in.

Note: if an $N \times P$ matrix is formed such that column i equals vector $\mathbf{d}_i$, then condition 2 says that no row has fewer than $\underline{L}$ 1s or more than $\overline{L}$ 1s, and condition 3 says that no column has fewer than $\underline{F}$ 1s or more than $\overline{F}$ 1s.

The cost function C is chosen so that minimizing it tends to minimize the components of l, f, and u. Minimizing the components of l alleviates the speed degradation caused by capacitive loading on the cells of Y. Minimizing f minimizes the fan-in (number of inputs) of the exclusive-OR gates whose outputs form the bit streams. Clearly, l and f are strongly correlated (minimizing the components of one tends to minimize those of the other). Minimizing the components of u tends to keep the bit streams uniformly separated in time. The exact form of the cost function C depends on the relative importance of minimizing these various quantities in a particular application.

If the loads on the cells of the shift register or the fan-ins of the exclusive-OR gates are of no concern, then the cost function C does not depend on l or f; moreover, $\underline{L}$ and $\underline{F}$ are small enough and $\overline{L}$ and $\overline{F}$ are large enough so that conditions 2 and 3 are satisfied trivially. The problem then reduces to generating P bit streams with specified, exact time separations. This problem has a simple analytical solution.

To see this, first note that the evolution of the shift register's state is governed by the following equation:

$$\begin{bmatrix} z_0^{k+1} \\ z_1^{k+1} \\ \cdot \\ \cdot \\ \cdot \\ z_{N-1}^{k+1} \end{bmatrix} = \mathbf{M} \begin{bmatrix} z_0^{k} \\ z_1^{k} \\ \cdot \\ \cdot \\ \cdot \\ z_{N-1}^{k} \end{bmatrix} \tag{6}$$

where the *state transition matrix* M is defined as follows:

$$\mathbf{M} \triangleq \begin{bmatrix} \mathbf{0}_{N-1} & \mathbf{I}_{(N-1)\times(N-1)} \\ c_0 \; c_1 & \cdots \; c_{N-1} \end{bmatrix} \tag{7}$$

Here $\mathbf{I}_{(N-1)\times(N-1)}$ denotes the $(N-1)\times(N-1)$ identity matrix, $\mathbf{0}_{N-1}$ denotes the $(N-1)$-component all-zero column vector, and the c_i are the feedback coefficients from Eq. (1).

Lemma 3 hereinbelow says that the taps for a given shift t are obtained explicitly by merely calculating the matrix $\mathbf{M}^t$ and inspecting its first row.

Lemma 3: Let Y denote a maximal LFSR initialized in a nonzero state and with state transition matrix M. Let t be a nonnegative integer. Then the tap coefficient vector d for Y that gives a shift forward in time by t clock cycles is the transpose of the first row of the matrix $\mathbf{M}^t$.

Proof: Let $\mathbf{z}^k$ denote the vector with components z_i^k (cf. Eq. (6)). Let $\mathbf{e}_0 \in \mathbf{F}^N$ denote the column vector with 1 as its zeroth component and 0s for the remaining $N-1$ components. Then the value of the PRBS generated by Y at clock cycle k is

$$a_k = \mathbf{e}_0^{\mathrm{T}} \mathbf{M}^k \mathbf{z}^0 \tag{8}$$

For any tap coefficient vector d, the output generated at clock cycle k is $\mathbf{d}^{\mathrm{T}} \mathbf{z}^k = \mathbf{d}^{\mathrm{T}} \mathbf{M}^k \mathbf{z}^0$. If $\mathbf{d}^{\mathrm{T}} = \mathbf{e}_0^{\mathrm{T}} \mathbf{M}^t$ is chosen, then the output at clock cycle k is $\mathbf{e}_0^{\mathrm{T}} \mathbf{M}^t \mathbf{M}^k \mathbf{z}^0 = \mathbf{e}_0^{\mathrm{T}} \mathbf{M}^{k+t} \mathbf{z}^0$. But this is the same as a_{k+t}, by Eq. (8). **Q.E.D.**

Lemma 3 provides a solution when the loads or fan-ins are of no concern. Note that $\mathbf{M}^t$ can be calculated in $\log(t)$ CPU time. One can calculate a table containing the matrix powers $\mathbf{M}^0, \mathbf{M}^1, \mathbf{M}^2, \mathbf{M}^4, \mathbf{M}^8, \cdots, \mathbf{M}^{2^{N-1}}$. Then the binary representation of t can be used to choose the powers of $\mathbf{M}$ to multiply together to calculate $\mathbf{M}^t$.

The previous special case showed that it is easy to calculate the taps necessary to obtain exact shifts when the load or fan-in are not a concern. When they are a consideration, the shifts must be allowed to vary from their nominal value (i.e., select a nonzero value for r) and a heuristic technique must be used to find a "good" set of taps. Since a fairly wide variance in the shift values can be allowed for this noise-generating application, solution candidates are abundant and a large state space may be searched to find a solution with acceptably low fan-in and cell load.

The software solution implemented for this problem can be described as follows. First, consider the set of tap patterns with K taps for an N-cell shift register. The number of such patterns is

$$\binom{N}{K} \triangleq \frac{N!}{K!(N-K)!} \tag{9}$$

The set of *essential* K-tap patterns is defined to be the smallest subset from which all K-tap patterns can be obtained by right-shifting a pattern from this subset by zero or more positions. When right-shifting a pattern, zeros are padded on the left. The set of essential patterns has only $\binom{N-1}{K-1}$ members, or K/N times the number of total patterns. For example, the number of 2-tap patterns for a 10-cell register is $\binom{10}{2} = 45$, while there are only $\binom{9}{1} = 9$ essential patterns, viz.,

1100000000
1010000000
1001000000
1000100000
1000010000
1000001000
1000000100
1000000010
1000000001

Note that once the bit stream shifts for the essential K-tap patterns are found, the bit stream shifts for all other K-tap patterns can be found

trivially. For example, if the shift of 1010000000 is q, then the shift of 0001010000 is $q - 3$ because the latter pattern is obtained from the former by right-shifting three bit positions.

Let X denote the collection of all essential tap coefficient vectors $\mathbf{d} \in \mathbf{F}^N$ with at least $\underline{F}$ 1s but not more than $\overline{F}$ 1s. The number of elements in X, $\|X\|$, is

$$\|X\| = \sum_{i=\underline{F}}^{i=\overline{F}} \binom{N-1}{i-1} \tag{10}$$

(This is a polynomial in N of order $\overline{F}-1$.) For each $\mathbf{d} \in X$, a record is stored in main memory that contains a representation of $\mathbf{d}$ along with the shift of its corresponding sequence (see hereinbelow regarding the calculation of this shift). Note that memory usage is greatly reduced by including only the essential tap patterns in the set X. Simulated annealing, or any desired random or deterministic technique, is used to search X to find a collection of tap coefficient vectors that minimizes the cost function and satisfies conditions 1 and 2 (condition 3 is satisfied by construction). If a solution exists, this method will find it given enough CPU time.

In practice, it was discovered that even the set of tap coefficient vectors with only two 1s produces shifts that are fairly well distributed throughout the interval $[0, 2^N - 2]$. Thus, the procedure is normally tried first with X containing just the tap coefficient vectors with $\underline{F}$ 1s. The members of X are bucket-sorted according to the nominal shift value to which they are closest. If all the buckets contain at least one tap, a solution is sought. If no satisfactory solution can be found, then the tap coefficient vectors with $\underline{F} + 1$ 1s are added to X and bucket-sorted, and the best solution is sought again. This process (of adding a new set of tap coefficient vectors to X then searching X for the best solution) is continued, if necessary, until the tap coefficient vectors with $\overline{F}$ 1s have been added to X.

Finding the shift associated with each $\mathbf{d} \in X$ can take significant CPU time. One straightforward way to do this is a method called *simple shifting*. Here an efficient representation $\mathbf{Y}$ of the shift register is implemented using the word operations of the host computer. For a given nonzero initial state $\mathbf{z}^0$ of the shift register, the first N bits of the sequence G corresponding to a given tap coefficient vector $\mathbf{d}$ can be calculated easily

using $\mathbf{Y}$. Let $\mathbf{g}^* \in \mathbf{F}^N$ denote the first N bits of G. Note that $\mathbf{g}^*$ represents the state of $\mathbf{Y}$ at the clock cycle that equals the shift of G. Thus, starting at the given initial state $\mathbf{z}^0$, $\mathbf{Y}$ is shifted one clock cycle at a time until its state is found to equal $\mathbf{g}^*$. The clock cycle where this equality occurs is the shift associated with $\mathbf{d}$.

The simple shifting method uses $O(1)$ (i.e., constant) memory and $O(2^N)$ CPU time. It can exact a large time penalty for practical problems. For example, a maximal 25-stage shift register has a sequence length of 34 million clock cycles. Thus, it would be expected that it would be necessary to shift $\mathbf{Y}$ an average of 17 million times for each $\mathbf{d} \in X$. In practice, however, it has been found that simple shifting is too slow for problems of "practical" size, i.e., when the shift register has more than about 20 cells.

Faster calculation at the expense of increased main memory usage can be obtained with a variant of what is known as Shanks' *giant step/baby step* method. (see, for example, D. E. Knuth, *The Art of Computer Programming, Vol. III: Sorting and Searching*. Reading, MA: Addison-Wesley, 1973, p. 9.) Here are stored a collection of bit patterns representing the states of the shift register at uniformly-spaced clock cycle intervals. Then given a tap coefficient vector $\mathbf{d}$, the associated shift register state $\mathbf{g}^*$ is calculated, as was done for the simple shifting method. The shift register representation $\mathbf{Y}$ is started in the state $\mathbf{g}^*$. It is then shifted one clock cycle at a time until its state is found to equal one of the bit patterns stored in the table. The shift associated with $\mathbf{d}$ is then the shift of the table bit pattern less the number of shifts needed to bring $\mathbf{Y}$ to that state.

In more detail, this method proceeds as follows. First, a "reasonable" *giant step* size h is chosen. As will be noted, a small h implies a fast calculation of the shift for each tap pattern, but the cost in memory usage and table setup time grows as h becomes smaller. Therefore, a compromise value of h must be chosen. For the example of a 25-stage shift register, $h = 5000$ might be chosen. Next, for all integers i such that $i \geq 0$ and $ih \leq 2^N - 2$, a hash table is filled with records, each containing the integer ih and the bit pattern $\mathbf{M}^{ih}\mathbf{z}^0$ for some specified nonzero initial state $\mathbf{z}^0$. For the example, this means that $v \triangleq \lfloor (2^N - 2)/h \rfloor + 1 = 6711$ bit patterns are calculated and installed in the hash table, where $\lfloor x \rfloor$ denotes the greatest integer not greater than x. If $\mathbf{E} = \mathbf{M}^h$ is initially calculated, then the hash table building takes v matrix multiplications. Once $\mathbf{w} = \mathbf{M}^{ih}\mathbf{z}^0$ has been

calculated for some value of i, $\mathbf{M}^{(i+1)h}\mathbf{z}^0$ is simply $\mathbf{Ew}$.

As in the case of the simple shifting method, let $\mathbf{Y}$ denote an efficient representation of the shift register, let $\mathbf{d}$ denote a tap pattern, and let $\mathbf{g}^*$ denote the first N bits of the bit stream corresponding to $\mathbf{d}$ when the initial state of $\mathbf{Y}$ is $\mathbf{z}^0$. To find the shift t associated with $\mathbf{d}$, $\mathbf{Y}$ is initialized to the state $\mathbf{g}^*$. Also, counter r is initialized to zero. Then t is found as follows:

1. Lookup the bit pattern that represents the state of $\mathbf{Y}$ in the hash table. If this bit pattern, which equals $\mathbf{M}^{ih}\mathbf{z}^0$ for some i, is in the hash table, set $t = ih - r$ and exit from loop; otherwise, go to step 2.
2. Shift $\mathbf{Y}$ by one clock cycle and increment counter r by 1. Go to step 1.

Note that the loop will never be executed more than h times, and, on average, it is executed $h/2$ times for each calculation of t. That is, the time complexity of each t calculation is $O(h)$. This results in a significant savings in CPU time for each t calculation relative to the simple shift method. Since $v \approx 2^N/h$ bit patterns must be stored in the hash table, the memory complexity in terms of N and h is $O(2^N/h)$. The time required to calculate the v bit patterns in the hash table is also proportional to v and is therefore $O(2^N/h)$. Clearly, the value of h must be chosen based on N and the number of t calculations to minimize the total time (setup time plus t calculation time) while keeping the memory usage within reasonable limits.

The tap-calculating procedure described above has been implemented in the C programming language (B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Prentice-Hall, Inc., 1978). For the shift registers of interest (i.e. those having fewer than 30 stages), it was found that the giant step/baby step algorithm was adequate for tap shift calculations. Even after the floating-point intensive code for tap cost calculation was optimized for efficient execution on a vector processor machine, it was found that the CPU time bottleneck was the solution search (optimization) step, *not* the tap shift calculation. For shift register larger than 30 stages, other algorithms may be needed for tap shift calculation.

A listing of the program appears in APPENDIX A. The user inputs the number of cells in the register, the feedback pattern, and the number of bit streams required. Also input is the maximum and minimum allowable loading on a cell, the maximum and minimum allowed fan-in, and the maximum allowed shift variation. In minimizing the cost function C, weighting factors are assigned to the $\mathbf{u}$, $\mathbf{l}$, and $\mathbf{f}$ components,

which are also specified by the user. In addition the user specifies the coefficients of a penalty function used when a potential solution falls outside the specified ranges.

The program was used to derive the tap patterns for 32 bit streams as generated from a 25-stage shift register. Since a 25-stage shift register produces a PRBS of maximal length 33,554,432 ($2^{25} - 1$), the time separation between bit streams is approximately one million clock cycles. The solution is shown in TABLE I. This solution search was run with the maximum and minimum fan-in set equal to three ($\underline{F} = \overline{F} = 3$). The minimum cell load ($\underline{L}$) was set at three and the maximum cell load ($\overline{L}$) was set at four. The maximum allowed shift variation was 0.4. The resulting solution had an average load per shift register cell of 3.8, with four cells having three connections and 21 cells having four connections. The actual maximum shift variation for this solution (maximum u_i from condition 1 of problem statement hereinabove) was 0.32. Each tap pattern line in TABLE I indicates the cells to be tapped to produce the bit stream having the particular shift. As an example, the first bit stream is generated from the modulo 2 combination from the outputs of the cells 9, 13, and 21, the cells being numbered from 0 to 24 from right to left, as hereinabove.

TABLE I
Tap Pattern Solution

number of bit streams: 32
feedback cells: 0 and 3
feedback pattern: 00000000000000000000001001
sequence length: 33,554,431
fan-in = 3, all bit streams
maximum number of taps on a cell: 4
minimum number of taps on a cell: 3
average number of taps per cell: 3.8
maximum shift variation: 0.32

Tap Pattern	Shift
00010000000100001000000000	813
00001000000000001000100000	977154
00000001000000000101000000	1918423
10000000010000000001000000	3065848
00000000001000000110000000	4202921
00010100000000010000000000	5199153
00000000001000000010000001	6561452
10000000000000001000000001	7319152
00000010000000000001010000	8405832
00000000001100000010000000	9153942
00000000010010000000100000	10119558
00000010000000000000000110	11177548
01000000100000000001000000	12240651
00000001100001000000000000	13588669
00000000000000010000000110	14519726
00100000000100000000001000	15641663
00000100000000000100100000	16777233
11000000000000000000001000	17769307
00010100000000000000000100	18739334
10000000100000000000000001	19774592
00001010000001000000000000	20796598
00001001000000000000000010	22115686
00000001000000000100000010	23150710
00000000010010000000010000	24450889
00000010000001000010000000	25165828
00000000000001100000000100	26245895
00101000000000010000000000	27177326
01000000000100010000000000	28164123
00100000000000010000100000	29180947
00100000100001000000000000	29900123
01000000010000000000001000	31268791
00010100000000000000010000	32533378
44444444443444444444433443	loading pattern

By using the techniques of the present invention in a CMOS implementation of a neural network to generate plural uncorrelated analog noise sources for annealing, the substantial cost in silicon area of the LFSR can be amortized over many neurons while the incremental cost per neuron is limited to some simple combinatorial logic. The function of the low-pass filter could also be served by the frequency response of the neuron itself, thereby saving the area cost associated with the filter. In addition to the area advantage, a single LFSR avoids the control and synchronization problems of multiple LFSRs.

In the example of TABLE I, the maximal length sequence becomes 32 separate bit streams with an average separation of about one million clock cycles. By clocking the LFSR at 100 MHz, each bit stream is separated from repetition by its nearest neighbor by approximately 10 milliseconds. Each bit stream is low pass filtered to about 5 MHz. An anneal cycle of 10 microseconds would therefore have about 50 analog zero crossings. In a network containing 32 neurons, each neuron would see noise that would not be repeated anywhere else in the network for about 1000 anneal cycles which is a substantially greater separation than is required. This same LFSR could conceivably be used for 1000 times as many neurons. For neural network applications, therefore, shift spacing is less important for design than fan-in or fan-out considerations.

The hardware advantage of the present invention is particularly important when such large numbers of bit streams need to be generated. In the present invention, for a given relative shift spacing between the bit streams, the number of cells in the shift register grows as $\log(P)$, where P is the number of bit streams. In contrast, the hardware requirement for prior art methods grows directly with P. In future generations of neural network chips, it is envisioned that hundreds and perhaps thousands of bit streams will be required. Accordingly, the hardware advantage of the present invention will be significant.

Although described in connection with providing noise sources for stochastic neural networks, the present invention has other applications. For example, bit-error rate testers use a pseudo-random bit stream to test communication systems at high speed. The speed is limited by the rate at which the shift register can be clocked. By providing multiple uncorrelated noise sources and then multiplexing them, a new pseudo-random bit stream at higher speed can be provided because a multiplexer

can operate faster than a shift register in a given technology. Alternatively, the bit-error rate tester can provide multiple outputs for parallel testing, which is generally not available in currently available equipment .

The above-described embodiment is illustrative of the principles of the present invention. Other embodiments could be devised by those skilled in the art without departing from the spirit and scope of the present invention.

What is claimed is:

1. A generator of plural pseudo-random bit streams comprising

a single maximal length linear feedback shift register having a plurality of cells;

for each bit stream, means for modulo 2 combining the tapped outputs of selected ones of said cells to produce the bit stream, the cell outputs selected to be tapped and combined being determined so that the bit streams are separated by predetermined shifts.

2. A generator of plural pseudo-random bit streams in accordance with claim 1 further comprising means for low-pass filtering each of the plural bit streams to produce plural sources of essentially Gaussian noise.

3. A generator of plural pseudo-random bit streams comprising

a single maximal length linear feedback shift register having a plurality of cells;

for each bit stream, means for modulo 2 combining the tapped outputs of selected ones of said cells to produce the bit stream, the cell outputs selected to be tapped and combined being determined so that the maximum allowed shift variation between bits streams, the maximum and minimum allowed fan-in, and the maximum and minimum cell load are within predetermined limits.

4. A generator of plural pseudo-random bit streams in accordance with claim 3 further comprising means for low-pass filtering each of the plural bit streams to produce plural sources of essentially Gaussian noise.

5. A stochastic element for a neural network comprising

a single maximal length linear feedback shift register having a plurality of cells;

means for producing plural pseudo-random bit streams from said single shift register by modulo 2 combining for each bit stream the tapped outputs of selected ones of said cells, the cell outputs selected to be tapped and combined being determined so that the bit streams are separated by predetermined shifts.

6. A stochastic element for a neural network in accordance with claim 5 further comprising means for low-pass filtering each of the plural bit streams to produce plural sources of essentially Gaussian noise.

7. A stochastic element for a neural network comprising

a single maximal length linear feedback shift register having a plurality of cells;

means for producing plural pseudo-random bit streams from said single shift register by modulo 2 combining for each bit stream the tapped outputs of selected ones of said cells, the cell outputs selected to be tapped and combined being determined so that the maximum allowed shift variation between bits streams, the maximum and minimum allowed fan-in, and the maximum and minimum cell load are within predetermined limits.

8. A stochastic element for a neural network in accordance with claim 7 further comprising means for low-pass filtering each of the plural bit streams to produce plural sources of essentially Gaussian noise.

FIG. 1
(PRIOR ART)

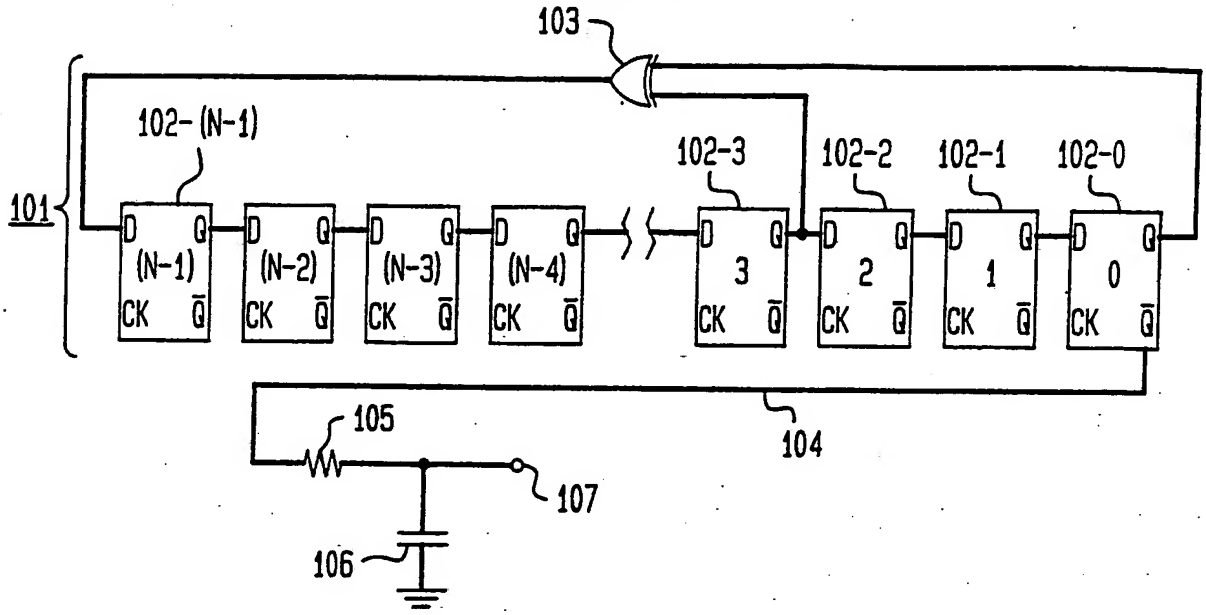
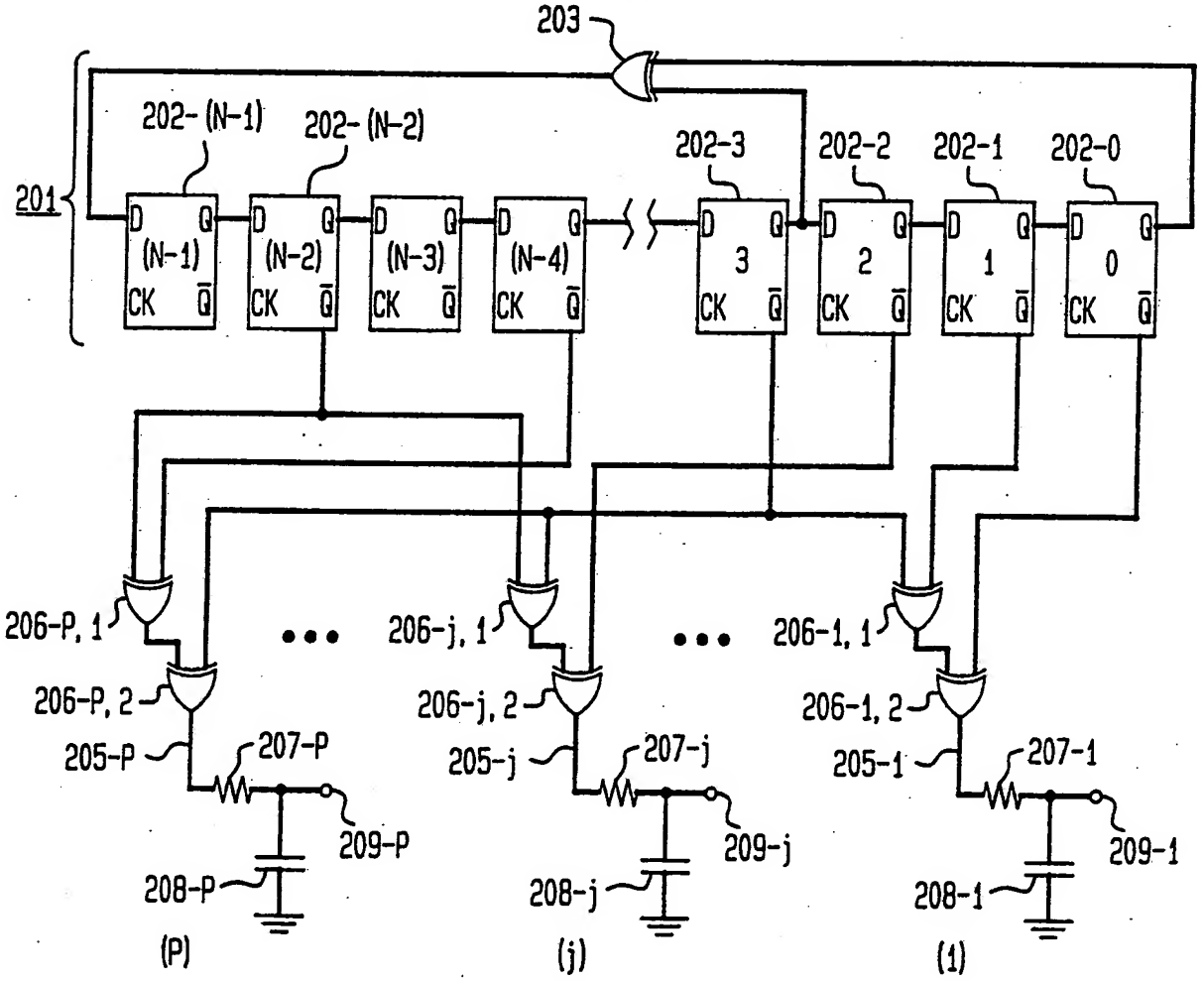


FIG. 2



INTERNATIONAL SEARCH REPORT

International Application No PCT/US90/04407

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC(5): G06F 1/02
US CL.: 364/717

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
US	364/717; 331/78

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [6]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
X Y	US, A, 3,881,099 (AILETT ET AL.) 29 April 1975 See the entire document.	1,3,5 & 7
A	US, A, 3,811,038 (REDDAWAY) 14 May 1974 See the entire document.	1-8
A	US, A, 4,325,129 (GROTH, JR.) 13 April 1982 See the entire document.	1-8
A	US, A, 4,748,576 (BEKER ET AL.) 31 May 1988 See the entire document.	1-8

* Special categories of cited documents: [15]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
18 OCTOBER 1990	11 FEB 1991
International Searching Authority [1]	**Signature of Authorized Officer [20]**
ISA/US	for André Robinson D. H. MALZAHN

Form PCT/ISA/210 (second sheet) (May 1986)